Parametric condition for single mouse monitoring

```
% Nastavení limitů parametrů

%Autocorrelation limits
autocorr_low = 0.11;
autocorr_high = 0.2;

% Piezosenzors amplitude limits
piezo_max = 0.3;
piezo_max_high = 0.5;

piezo_min = -0.3;
piezo_min_low = -0.5;

piezo_diff = 0.4;
piezo_diff_high = 0.9;

%Middle frequencies power
power= -85;
power_high = -70;

%Dominant frequency power
freq_high = 55;
freq_low = 30;

%Microphone amplitude and frequency
mic_max = 0.8;
mic_max_high = 1;

mic_min = -0.8;
mic_min_low = -1.1;

mic_diff = 1.6;
mic_diff_high = 1.9;

mic_freq = 200;



%Set path to directory with "framing" function
addpath('D:\Škola\ČVUT\Diplomka\Matlab\Records');

%Do not change this variables
start = 1;
index = 0;
false_p = 0;
% multiparam = [];
```

```
S = dir(fullfile('*.mat'));
S = struct2table(S);
[~,idx] = sort([S.date]);
names = string(S.name(idx));
len = length(names)-1;
clear S

for l = start:2:len

    %% Načítání dat avýpis kontrolovaného segmentu
    fprintf("Currently checking segment: %d\n",l-index);
    filename = names(l);
    piezo = load(filename);
    piezo = piezo.piezo;
    filename = names(l+1);
    mic = load(filename);
    mic = mic.mic;

    piezo1 = [piezo.Dev1_ai7, piezo.Dev1_ai4, piezo.Dev1_ai5, piezo.Dev1_ai6,
piezo.Dev1_ai0];
    t = seconds(piezo.Time);
    p1 = piezo1(:,1);
    p2 = piezo1(:,2);
    p3 = piezo1(:,3);
    p4 = piezo1(:,4);
    pulse = piezo1(:,5);

    fs = 4000;

    p1 = p1 - mean(p1);
    p2 = p2 - mean(p2);
    p3 = p3 - mean(p3);
    p4 = p4 - mean(p4);


    mici = [mic.Dev2_ai0];
    mic1 = mici(:,1);
    t_mic = seconds(mic.Time);
    fs_mic = 30000;
    mic1 = mic1 - 1.25;
    mic1 = highpass(mic1,125,fs_mic);

    sec = 90;

    frlen = sec*fs;

    frlen = cast(frlen,"uint32");
    frlen = cast(frlen,"double");
```

```
%% Segmentace signálu
hop = round(frlen*(1/2));
[FRM_p1, tfrm] = framing(p1, frlen, hop, fs);
[FRM_p2, ~] = framing(p2, frlen, hop, fs);
[FRM_p3, ~] = framing(p3, frlen, hop, fs);
[FRM_p4, ~] = framing(p4, frlen, hop, fs);


frlen = sec*fs_mic;

frlen = cast(frlen,"uint32");
frlen = cast(frlen,"double");

hop = round(frlen*(1/2));
[FRM_mic, tfrm] = framing(mic1, frlen, hop, fs_mic);
params = zeros(length(FRM_p1(1,:)),7);



%% Autokorelační funkce
score = 0;
for j = 1:length(FRM_p1(1,:))
    for i = 1:4
        if i==1
            sig = FRM_p1(:,j);
        end
        if i==2
            sig = FRM_p2(:,j);
        end
        if i==3
            sig = FRM_p3(:,j);
        end
        if i==4
            sig = FRM_p4(:,j);
        end
        [acf,lags] = autocorr(sig,NumLags=180);
        acf=-acf;
        [pks,~] = findpeaks(acf);
        if length(pks)<2
            continue
        end
        peak_y=pks(1);
        peak_y2=pks(2);
        peak_diff = abs(peak_y - peak_y2);

        if (peak_diff > autocorr_low) && (peak_diff < autocorr_high)
            score = score +1;
        end
    end
    if (score >= 2)
```

```
        params(j,1) = params(j,1) +2;
    end
    params(j,3) = score;
    score = 0;
end

    %% Amplituda z piezo-měničů
score=0;
for j=1:length(FRM_p1(1,:))
    for i=1:4
        if i==1
            sig = FRM_p1(:,j);
        end
        if i==2
            sig = FRM_p2(:,j);
        end
        if i==3
            sig = FRM_p3(:,j);
        end
        if i==4
            sig = FRM_p4(:,j);
        end
[ma,max_idx] = max(sig);
[mi,min_idx] = min(sig);

if (max_idx+fs) > length(sig)
    max_sig=sig(max_idx-fs:end);
elseif (max_idx-fs <= 0)
    max_sig=sig(1:max_idx+fs);
else
    max_sig=sig(max_idx-fs:max_idx+fs);
end

if (min_idx+fs) > length(sig)
    min_sig=sig(min_idx-fs:end);
elseif (min_idx-fs <= 0)
    min_sig=sig(1:min_idx+fs);
else
    min_sig=sig(min_idx-fs:min_idx+fs);
end

min_maxsig = min(max_sig);
max_minsig = max(min_sig);

max_diff = abs(ma - min_maxsig);
min_diff = abs(mi - max_minsig);

if (max_diff > min_diff)
    diffi = max_diff;
    mi = min_maxsig;
else
    diffi = min_diff;
    ma = max_minsig;
end

    if (ma > piezo_max)
        score = score +1;
      if (ma > piezo_max_high)
          score = score +2;
      end
    end

    if (mi <piezo_min)
    score = score +1;
      if (mi < piezo_min_low)
          score = score +2;
      end
    end
  if (diffi > piezo_diff)
      score = score +1;
  end
  if (diffi > piezo_diff_high)
      score = score +2;
  end
    if score>=2
        params(j,1) = params(j,1) +1;
        params(j,2) = params(j,2) + 1;
    end
    if score>=5
        params(j,1) = params(j,1) +2;
        params(j,2) = params(j,2) + 2;
    end

    score = 0;
    end

end


    %% Energie středních kmitočů
for j=1:length(FRM_p1(1,:))
    for i=1:4
        if i==1
            sig = FRM_p1(:,j);
        end
        if i==2
            sig = FRM_p2(:,j);
        end
        if i==3
            sig = FRM_p3(:,j);
    end
    if i==4
        sig = FRM_p4(:,j);
    end
        [p,f] = pspectrum(sig,fs);
        p = pow2db(p);
        idx = find(f>500,1);
        pow = max(p(idx:end));

    if pow>power
        params(j,1) = params(j,1) +1;
        params(j,5) = params(j,5) + 1;
    end
    if pow>power_high
        params(j,1) = params(j,1) +2;
        params(j,5) = params(j,5) + 2;
    end
end
end


%% Dominantní energie v segmentu
score = 0;
for j=1:length(FRM_p1(1,:))
for i=1:4
    if i==1
        sig = FRM_p1(:,j);
    end
    if i==2
        sig = FRM_p2(:,j);
    end
    if i==3
        sig = FRM_p3(:,j);
    end
    if i==4
        sig = FRM_p4(:,j);
    end
    y = fft(sig);

    n = length(sig);
    f = (0:n-1)*(fs/n);
    powery = abs(y).^2/n;

    [~,idx]=max(powery);

    if (f(idx) <= freq_high) && (f(idx) >= freq_low)
        score = score + 1;
    end
end
if score>=3
```

```
            params(j,1) = params(j,1) +2;
        end
        params(j,4) = score;
        score = 0;
    end

    %% Vyhodnocení signálu z mikrofonu
p = min(length(FRM_p1(1,:)),length(FRM_mic(1,:)));
score = 0;
for i=1:p
    sig = FRM_mic(:,i);

    [ma,max_idx] = max(sig);
    [mi,min_idx] = min(sig);

    if (max_idx+(fs_mic/2)) > length(sig)
        max_sig=sig((max_idx-(fs_mic/2)):end);
    elseif (max_idx-(fs_mic/2) <= 0)
        max_sig=sig(1:max_idx+(fs_mic/2));
    else
        max_sig=sig(max_idx-(fs_mic/2):max_idx+(fs_mic/2));
    end

    if (min_idx+(fs_mic/2)) > length(sig)
        min_sig=sig(min_idx-(fs_mic/2):end);
    elseif (min_idx-(fs_mic/2) <= 0)
        min_sig=sig(1:min_idx+(fs_mic/2));
    else
        min_sig=sig(min_idx-(fs_mic/2):min_idx+(fs_mic/2));
    end

    min_maxsig = min(max_sig);
    max_minsig = max(min_sig);

    max_diff = abs(ma - min_maxsig);
    min_diff = abs(mi - max_minsig);

    if (max_diff > min_diff)
        mic_diffi = max_diff;
        mi = min_maxsig;
    else
        mic_diffi = min_diff;
        ma = max_minsig;
    end

    if ma > mic_max
        params(i,1) = params(i,1) +1;
    end

    if ma > mic_max_high
```

```
        params(i,1) = params(i,1) +2;
        params(i,6) = params(i,6) + 2;
    end

    if mi < mic_min
        params(i,1) = params(i,1) +1;
        params(i,6) = params(i,6) + 1;
    end

    if mi < mic_min_low
        params(i,1) = params(i,1) +2;
        params(i,6) = params(i,6) + 2;
    end

    if mic_diffi > mic_diff
        params(i,1) = params(i,1) +1;
        params(i,6) = params(i,6) + 1;
    end

    if mic_diffi > mic_diff_high
        params(i,1) = params(i,1) +2;
        params(i,6) = params(i,6) + 2;
    end
   score = 0;

    y = fft(FRM_mic(:,i));
    n = length(y);
    f = (0:n-1)*fs_mic/n;
    power_mic = abs(y).^2/n;
    [~,idx]=max(power_mic);
    freq = f(idx);
    if (freq < mic_freq)
        params(i,1) = params(i,1) +1;
    end
    params(i,7) = freq;
end

possible_seizure = (find(params(:,1)>=10 & params(:,1)<17))';
seizure = (find(params(:,1)>=17))';
Max_score = max(params(:,1))



%% Vyhodnocení výsledků a výpis dat v tabulce
    if length(possible_seizure)>=1 || length(seizure)>=1
        false_p = false_p + 1; %Počet falešně pozitivních segmentů
        times = zeros(length(possible_seizure),2);
        times(:,1)=(tfrm(possible_seizure)-(sec/2))/60;
        times(:,2)=(tfrm(possible_seizure)+(sec/2))/60;
          disp(l-index);
          disp(names(l));
          disp('Possible seizure:')
          disp(possible_seizure);
          disp('Time of possible seizures');
          disp(times);
          % disp(tab);
          if length(seizure) >= 1
          disp('Seizure:')
          disp(seizure)
          times = zeros(length(seizure),2);
          times(:,1)=(tfrm(seizure)-(sec/2))/60;
          times(:,2)=(tfrm(seizure)+(sec/2))/60;
          disp('Time of seizures');
          disp(times);
          end
          % Tabulka s jednotlivými parametry a vypočteným skóre
          types=["Score","Amplitude",
"Autocorr","Dominant_freq","Freq_energy","Mic_amplitude","Mic_frequency"];
          tab =
table(params(:,1),params(:,2),params(:,3),params(:,4),params(:,5),params(:,6),param
s(:,7),VariableNames=types);
        start = l+2;
        index = index + 1;
        break;
    end
    index = index + 1;
end

if l==len && isempty(possible_seizure)
    disp('Done')
end
```